Nyelvi hasonlóságon alapuló intelligens keresés fordítómemóriában

Hodász Gábor

Pázmány Péter Katolikus Egyetem
Információs Technológiai Kar
H-1083 Budapest, Práter utca 50/a.
hodasz@morphologic.hu

Kivonat. A cikkben bemutatásra kerül a nyelvi hasonlóság egy definíciója, amely elméleti alapját képezi a nyelvi hasonlóságon alapuló intelligens keresésnek. A definíció lehetőséget ad a hasonlóság Levenshtein-távolság alapú vizsgálatára több nyelvi szinten. Az általános definíción túl bemutatom a fejlesztés alatt álló alkalmazást, amely 3 nyelvi szinten alkalmazza a távolságot: a szavak felszíni alakja szerint, tövesített alakjuk szerint és szófajuk szerint. A példákban így két mondat között egy 3 elemű vektor írja le a hasonlóságot. Bemutatom az elkészült teszt-környezet is, amely szövegfájlban keres egy adott jelölt-mondathoz a definíció szerint hasonló mondatokat. Végül vázolom a további munkákat és terveket.

1 Bevezető

A bemutatott nyelvi hasonlóság definíciójának célja az intelligens keresés megvalósítása a fordítómemóriában. A felhasznált keretrendszer a MorphoLogic Kft-nél fejlesztett MorphoTM fordítómemória, amely a MetaMorpho szabály-alapú fordítástámogató rendszer alapjain, az ott kidolgozott szabály-szintaxist alkalmazza.

A MorphoTM rendszer olyan fordítástámogató eszköz, amelynek célja, hogy a hagyományos fordítómemória-funkciókat nyelvi intelligenciával kiegészítve a jelenlegi rendszereknél többször ajánljon fordítást, és azok jobban közelítsék a kívánt minőségű fordítást. A fordítás egységei a mondatnál kisebb szegmensek (főnévi szerkezetek és az ezeket tartalmazó mondatvázak), amelyeket a forrás- és célnyelvi elemzők állítanak elő. Az adott bemeneti mondathoz hasonló szegmenseket „nyelvi intelligencián" alapuló távolság segítségével keressük, és a megszülető új fordításokat mint szabályokat tároljuk, amelyek a gépi fordítás minőségét folyamatosan javítják.

2 A fordítómemória működésének leírása

2.1 Szabály és minta

A MorphoTM fordítómemória egyesíti a szabály-alapú gépi fordítás és a minta alapú, statisztikus megközelítés előnyeit. A tárolt minták a MorphoLogic MetaMorpho nevű gépi fordítórendszerének formalizmusát követik: az egyes szabályok forrás és célnyelvi részből állnak, azaz minden szabály önállóan tartalmaz egy nyelvi elemet és annak fordítását. A szabályrendszer jellemzője, hogy homogén: nem különböztetünk meg lexikon-szerű és szintaxis-szerű szabályokat [Prószéky96, Prószéky02].

A fenti két tulajdonság lehetővé teszi, hogy a fordítómemóriába kerülő szabályok bármilyen szintű nyelvtani struktúrát leírhatnak, legyen az egyetlen főnév és fordítása, vagy egy mondatváz, amelyben üres helyek jelzik a főnevek helyét és a vonatkozó megkötéseket. Így a szabályok egyben nyelvi minták is, a szabálybázis pedig tekinthető speciális párhuzamos korpusznak is.

A szabályok az elemzés-fordítás folyamán egyszerű unifikációs nyelvtan szerint működnek. Az egyes szabályokban a különböző jegyek (megkötések) határozzák meg a szabály specifikusságát. A szabályban a nem kitöltött megkötések a konkrét mondat elemzése során kerülnek kitöltésre. Így a fordítómemória működése folyamán egy korábban eltárolt minta akkor is releváns lehet az aktuális mondat fordításában, ha előzőleg más morfológiai jegyekkel szerepelt. Ehhez az szükséges, hogy a szabályok eltárolásakor meghatározzuk a kellő megszorításokat, a többi jegyet azonban kitöltetlenül hagyjuk. Így a fordítómemória által megtalált korábbi fordítás csak abban az esetben lesz jelölt, ha kielégíti a szükséges megszorításokat. A nem szükséges megszorítások (pl. szám, személy, idő stb.) pedig a célnyelvi mondat generálása során az aktuális forrásnyelvi megfelelőik szerint kerülnek kitöltésre. Ez a megközelítés lehetővé teszi, hogy a hagyományos fordítómemóriával szemben például az angol 'go' igének különböző idejű alakjait (pl. 'went', 'has gone' stb.) annak ellenére megtalálja a rendszer a minták között, hogy közöttük a karakter-alapú távolság igen nagy.

2.2 Bővítés és fordítás

A MorphoTM alapvetően fordítómemóriaként működik, azaz fejlett eszközökkel támogatja az emberi fordítót, valamint lehetőséget ad az adatbázis bővítésére. A fordítói munka során a fordított mondatok feldolgozásával bővül a szabálybázis, és emellett lehetőség van minták párhuzamos korpuszból való automatikus felvételére is.

A fordítási folyamat során az emberi fordító felügyeli a fordítás folyamatát. A folyamat lépései vázlatosan a következők:

(1) A fordítómemória a berérkező mondatot megelemzi.
   a) Amennyiben ez sikeres, akkor előállnak a forrásnyelvi mondat szavainak lemmái, a mondat váza és a főnévi csoportok (*noun phrase*, NP).
   b) Amennyiben nem sikerül a teljes elemzés, úgy a rendszer megpróbálja egy sekély NP nyelvtannal elemezni a mondatot, így előállítva lemmákat, a vázat és az NP-ket.

c) Ha ez sem jár sikerrel, úgy az egész mondatot váznak tekintjük, amely nem tartalmaz főnévi csoportokat. Ez esetben a morfoszintaktikai elemző előállítja a mondat lemmáit.

(2) A lemmák sorozatát feldolgozzuk a fordítómemória igényeinek megfelően: meghatározzuk a szükséges morfológiai jegyeket, amelyek teljesülését megkívánjuk a tárolt minták közötti keresés során. A feldolgozás során külön kezeljük a speciális lemmákat (pl. Internet vagy e-mail cím, dátum, szám stb.), amelyek az esetlegesen szükséges módosítások után átkerülnek a célnyelvi mondatba.

(3) A főnévi csoportok (NP-k) és a többi lemmából álló mondatváz alapján az „intelligens" kereső hasonló mondatot, illetve hasonló NP-ket keres az adatbázisban.

a) azok a szabályok, amelyek forrásnyelvi oldala teljesen lefedi a mondatot, a mondatvázat vagy egy NP-t, „elsülnek".

b) amennyiben az elsült szabályok nem fedik le a teljes mondatot, úgy a hasonlósági kereső modul a definiált nyelvi hasonlóság szerint hasonló mondatvázakat, illetve NP-ket keres az adatbázisban.

(4) A találatokat megfelelően szűrve és rangsorolva előáll a célnyelvi szegmensfordítás-jelöltek listája.

(5) A jelöltekből, illetve az opcionálisan gépi fordítással előállított célnyelvi szegmensekből összeáll az eredeti mondat felajánlott fordítása, amely tartalmazza a speciális lemmákat is.

(6) A felajánlott fordítás(oka)t a felhasználó elfogadhatja vagy módosíthatja.

(7) A módosított célnyelvi szegmensek elemzése és forrásnyelvi párjukkal való eltárolásuk révén új szabályokkal bővül a fordítómemória. Amennyiben a főnévi szerkezetek szinkronizálása gépi úton nem megvalósítható, úgy a fordító javíthatja a felajánlott hozzárendeléseket.

3 A nyelvi szerkezetek közötti hasonlóság

Ebben a fejezetben leírását adjuk a nyelvi hasonlóságon alapuló keresés egy megközelítésének, amely segítségével a fordítómemóriában levő mondat-vázak és főnévi szerkezetek nyelvi hasonlóság alapján kereshetőek. Ehhez definiáljuk a nyelvi hasonlóság fogalmát.

3.1 Hasonló munkák

Bár a fordítómemóriák és a példa-alapú gépi fordítás elmélete [Nagao84] és alkalmazásai már a kilencvenes években megjelentek, a mondatok közötti hasonlósággal csak az évezredfordulón és utána, azaz napjainkban kezdtek foglalkozni a tudományos munkák. A nyelvi hasonlóság fogalmára azonban egészen a legutóbbi időkig tudtunkkal csak egy kutatócsoport adott definíciót [Mandreoli02]. A gyakorlati alkalmazásra azonban számtalan publikáció született, amelyek tekintélyes része a Levenshtein-távolság [Levenshtein65] kiterjesztésével és a dinamikus programozás eszközeivel ad megoldást a hasonlóság kezelésére [Planas00]. A példa-alapú gépi fordítás megvalósí-

tásához elengedhetetlen a minták hasonlóságának megfelelő kezelése, így a tárgyban született cikkek közül több foglalkozik a témával, illetve részproblémáival, mint a félszabad morfémák (function words) [Sumita93], a főnévi szerkezetek (terminológiák) [Sato93] és az elöljárós szerkezetek [Sumita93].

Az eddigi megközelítések legfontosabb hiányosságai:

- számos megoldás egyedi nyelvi ismereteket kíván, mint pl. az ekvivalenciaosztályok, vagy egyéb szemantikai tulajdonságok
- legtöbbször nem definiálnak nyelvi hasonlósági mértéket, így a találatok halmaza nem rangsorolható
- a legtöbb megközelítésben a mondat a keresés legkisebb egysége

A kereskedelmi forgalomban jelenleg kapható fordítás-támogató rendszerek legtöbbje még a fentebb vázoltaknál is kevesebb nyelvi alapú algoritmust alkalmaz. Hatékonyságukat egyedül gyorsaságuk és a fordítandó szövegek nagyfokú hasonlósága indokolja.

3.2 A nyelvi hasonlósági mértéke

A nyelvi hasonlóság mértékének definiálásakor a következő elvárásokat kell figyelembe venni:

- A hasonlóság mértéke legyen a lehető legnagyobb mértékben független a nyelvi környezettől, a szemantikai kontextustól. A fordítómemória kezeli a kontextust, annak meghatározását a fordítóra bízza, azonban a hasonlóság mértéke ettől független.
- A hasonlóság vegye figyelembe mind a szószintű különbségeket (törlés, beszúrás, csere), mind pedig a morfoszintaktikai jegyek különbségét. A mondatváz és az NP-k megkülönböztetésén kívül szintaktikai jegyeket jelenleg nem veszünk figyelembe.

A teljes mondatok, a mondatvázak és a főnévi csoportok olyan szimbólumok sorának tekinthetők, amelyek meghatározott jegyekkel rendelkeznek, amely jegyek közül van kitöltött és kitöltetlen. Mivel a lexikai jegy csak egy ezen jegyek közül, így minden szegmenst, akár teljes mondat, akár főnévi csoport, olyan szimbólumok sorának tekintjük, ahol a lexikai jegyek kitöltöttek, míg a mondatvázakban a főnévi csoportok helyét lexikai jegy nélküli szimbólumok jelzik, amelyek a fordítás során kitöltődhetnek. Ezért a továbbiakban azonos módon kezelhetünk minden szegmenst.

3.3 A hasonlóság szintjei

A hasonlósági távolság definíciójában a közismert Levenshtein-távolságot (szerkesztési távolság) [Levenshtein65] vesszük alapul. A nyelvi szerkezetek távolságának meghatározásakor figyelembe kell venni a nyelvi szerkezetek különböző szintjeit. Ebben a megközelítésben az m hosszú S szegmenst nem csupán szavak sorozatának tekintjük, hanem a különböző elemzési szinteknek megfelelően L szinten (layer) párhuzamosan összerendelt szimbólumok sorozatának, amely minden szinten m darab szimbólumot tartalmaz. Minden szinten az i-ik szimbólum egyértelműen megfeleltethető az S mondat i-ik szavának, és ebből a szóból különböző szintű nyelvi elemzés útján áll elő.

A MorphoTM rendszerben a következő hasonlósági szinteket definiáljuk:

- felszíni alak *(L1)*
- lemmatizált alak *(L2)*
- szófaj *(L3)*

Példa (1):

The fat mice eat cheese.

	T_1	T_2	T_3	T_4	T_5	T_6
L_1	The	fat	mice	eat	cheese	.
L_2	the	fat	mouse	eat	cheese	Punct
L_3	Det	Adj	Noun	Verb	Noun	Punct

Amennyiben a rendszer elő tudja állítani a mondat vázát és főnévi szerkezeteit, akkor a következő 2 NP- és 1 mondatváz-szegmens születik:

	T_1	T_2	T_3	T_4	T_5	T_6
L_1	The	fat	mice	eat	cheese	.
L_2	the	fat	mouse	eat	cheese	Punct
L_3	Det	Adj	Noun	Verb	Noun	Punct
$k(S_1)$	**NP_1**			**eat**	**NP_2**	

Bár a MorphoTM rendszerben (amint a fenti példákban is látható) a hasonlóságot 3 szinten számítjuk, a definíció lehetővé teszi akár több, akár kevesebb szint alkalmazását is. További szinteken más nyelvi (pl. szemantikai) vagy nem nyelvi (pl. formázási) információ is feldolgozható. Azonban minden egyes szint növelheti a számítás időigényét, amely a nyelvi elemzések többértelműsége, az elemzések bonyolultsága miatt nagyságrendekkel növelheti a feldolgozási időt, valamint további többértelműségi problémákat vethet fel.

Már a fenti példában is, de a mondatok túlnyomó részében a különböző elemzési szinteken fellép a többértelműség problémája. A példamondat 'fat' szava egyaránt lehet melléknév ('kövér, zsíros, hájas'), főnév ('kövérség, zsír, háj') vagy ige ('hízik, hízlal'). A helyes elemzés eldöntése szintaktikai elemzést kíván, amely viszont nincs benne az általunk használt 3 szintben. A szintaktikai elemzés bevezetése, bár elméletileg illeszkedik a vázolt definícióba, azonban olyan bonyolultságú nyelvi problémákat hozna a jelen konkrét megvalósításunkba, amely sem feldolgozási időben sem nyelvi erőforrásban (szabály-bázis méretében) nem megfelelő. A többértelműség kezelésének lehetséges módjait később mutatom be.

4 Levenshtein-távolságon alapuló hasonlóság

4.1 A távolság definíciója

(1) Definíció (Szerkesztési távolság szimbólumok sorozatára). *Legyen S_1 és S_2 két szegmens, amely a következő nyelvi szimbólumokból áll: $\sigma(S_1)=t_1^1..t_n^1$ és $\sigma(S_2)=t_1^2..t_m^2$. A szerkesztési távolság $\sigma(S_1)$ és $\sigma(S_2)$ között $(ed(\sigma(S_1),\sigma(S_2)))$ az a minimális műveletigény (beszúrások, törlések és helyettesítések száma), amely $\sigma(S_1)$-t $\sigma(S_2)$-be viszi.*

A fenti definíció a reprezentáció bármilyen szintjén alkalmazható. Ha a felszíni alakból nyelvi elemzési lépések tetszőleges sorával egy új hasonlósági szint áll elő, akkor az előálló szimbólumok sorát $\phi(S)$-sel jelölve $ed(\phi(S_1), \phi(S_2))$ a fentieknek megfelelően definiálható és számítható. A teljes hasonlóságot az összes hasonlósági szinten kiszámított távolság-értékekből képzett vektorként definiáljuk.

(2) Definíció (Több-rétegű szerkesztési távolság nyelvi szimbólumok sorozatára). *Legyen S_1 és S_2 két szegmens, mely a $\sigma(S_1)=t_1^1..t_n^1$ és $\sigma(S_2)=t_1^2..t_m^2$ nyelvi szimbólumaiból a $\phi_1,...,\phi_i$ nyelvi elemzési lépések segítségével képzett i darab hasonlósági réteget tartalmaz: $L_i,...L_1$. A több-rétegű szerkesztési távolság $\sigma(S_1)$ és $\sigma(S_2)$ között*

$$ED(\sigma(S_1),\sigma(S_2))=$$

$$[ed_{Li}(\sigma(S_{1,Li}),\sigma(S_{2,Li})),ed_{Li-1}(\sigma(S_{1,Li-1}),\sigma(S_{2,Li-1})),...,ed_{L1}(\sigma(S_{1,L1}),\sigma(S_{2,L1}))]$$

A (2) definícióban meghatározott hasonlósági vektor intelligens hasonlóságkeresésre megfelelő, és a későbbiekben látni fogjuk, hogy a hatékonyság növelése érdekében számítási egyszerűsítéseket alkalmazhatunk. Az egyes hasonlósági rétegeken a távolság-érték kiszámításának módja a Levenshtein-távolság számításának hagyományos, dinamikus programozás szerinti algoritmusa [Wagner&Fischer74].

Konkrét alkalmazásunkban a szerkesztési távolság helyett a távolság és a fordítandó mondat hosszának arányával számolunk:

$$d(\sigma(S_1),\sigma(S_2))=\frac{ed(\sigma(S_1),\sigma(S_2))}{|\sigma(S_1)|}$$

Illetve a hasonlósági vektor ennek megfelelően:

$$D(\sigma(S_1),\sigma(S_2))=\left[\frac{ed_{Li}(\sigma(S_{1,Li}),\sigma(S_{2,Li}))}{|\sigma(S_{1,Li})|},...,\frac{ed_{L1}(\sigma(S_{1,L1}),\sigma(S_{2,L1}))}{|\sigma(S_{1,L1})|}\right]$$

Példa (2):

S_1: *The fat mice eat cheese.*

S_2: *Cats eat mice.*

S_1:

	T_1	T_2	T_3	T_4	T_5	T_6
L_1	The	fat	mice	eat	cheese	.
L_2	the	fat	mouse	eat	cheese	Punct
L_3	Det	Adj	Noun	Verb	Noun	Punct

S_2:

	T_1	T_2	T_3	T_4
L_1	Cats	eat	mice	.
L_2	cat	eat	mouse	Punct
L_3	Noun	Verb	Noun	Punct

A teljes mondatokra értelmezett hasonlósági vektor a következőképpen fog alakulni:

$ed_{L3}(\sigma(S_{1,L3}),\sigma(S_{2,L3}))=2$ (2 törlés)

$ed_{L2}(\sigma(S_{1,L2}),\sigma(S_{2,L2}))=4$ (2 törlés és 2 helyettesítés)

$ed_{L1}(\sigma(S_{1,L1}),\sigma(S_{2,L1}))=4$ (2 törlés és 2 helyettesítés)

Így a hasonlósági vektor a következő lesz:

$ED(\sigma(S_1),\sigma(S_2))=[2,4,4]$

A mondatok hosszával osztott hasonlóság:

$D(\sigma(S_1),\sigma(S_2))=[0.33,0.66,0.66]$

Amint az a vektor számításának definíciójából is látszik, az egyes értékek között szoros összefüggés van, azaz a vektor értékei nem függetlenek egymástól. Abban az esetben, ha a szónak nincs többféle elemzése (nincs több szótöve és/vagy szófaja), akkor a felszíni alak egyezése maga után vonja az összes többi réteg egyezését is. Valamint megfordítva: (ugyanebben a meglehetősen ritka esteben) a szófaj-szinten (L_3) való különbözőség szinte biztosan különbözőségre vezet a fentebbi rétegeken.

Amennyiben (és az esetek túlnyomó részében ez az igaz) egy adott szónak több elemzése van, úgy kezelnünk kell a többértelműségből fakadó problémát. Amennyiben rendelkezünk olyan modullal, amely szintaktikai elemzés nélkül, egyéb (pl. statisztikai) úton tud szófaj szinten egyértelműsíteni (POS-tagging), akkor támaszkodhatunk ezen modul által egyértelműsített elemzésre. Amennyiben ilyen modult nem tudunk/akarunk alkalmazni, akkor a keresés algoritmusán kell változtatnunk. Erre két lehetőségünk kínálkozik:

- „vízszintes egyezés-keresés": az adott szegmens összes szavának összes elemzésében megpróbálunk olyan utat találni, amelyik leginkább hasonlít a kereső-mondathoz. A lehetséges utak közül a legkisebb távolságút fogadjuk el a két mondat távolságának az adott szinten.
- „függőleges egyezés-keresés": Az egyes szavakat mintegy különálló egységként kezelve, csak az adott szó lehetséges elemzései között keresünk. Ha van az adott szónak egyező elemzése a kereső mondat adott szavával, akkor elfogadjuk azon a szinten egyezőnek a két szót.

A MetaMorphoTM rendszerben rendelkezésre áll egy szófaji egyértelműsítő (POS-tagger), így a rendszer jelen állapotában a POS-tagger által ajánlott elemzést veszem alapul, a többi elemzést elhagyom. A rendszer következő verziójában már a fent vázolt mindkét egyezés-keresést megvalósítom tesztelés céljából.

A hasonlósági vektor számítási menete mindig az absztrakt szintek felől a felszíni szintek felé (L_3-tól L_1 felé) halad, amely a jövőben beépítendő egyszerűsítő algoritmusokra ad lehetőséget. Amennyiben az absztrakt szinten a távolság túl nagy (adott küszöbérték feletti), úgy nem érdemes a további szinteket vizsgálni.

5 Előnyök

A fenti példából jól látszanak a módszer előnyei:

- a nyelvi elemzés segítségével kiszámolt hasonlóság közel áll ahhoz, amit az emberi fordító is hasonlónak érez (a példában: 'X eat Y.')
- a különböző felszíni alakok közötti eltérések nem befolyásolják a szükségesnél jelentékenyebben a hasonló mondatok megtalálását (pl. közel kerülnek egymáshoz az angol 'go' ige különböző alakjai: 'went', 'has gone' stb.)
- A mondatvázak és a főnévi csoportok külön kezelésével lehetőség nyílik a mondatnál kisebb egységek automatikus fordítására is, még akkor is, ha nem találunk a mondatvázhoz hasonlító mondatot a memóriában.
- A szintek egymás utáni vizsgálatával akár már az első összehasonlításnál (a szófajok vizsgálatánál) eldönthető, hogy folytassa-e az algoritmus az összehasonlítást.

6 A módszer hátrányai és megoldandó feladatai

- Minden olyan algoritmus, amely nyelvi elemzést használ, a következő problémákat hozza be a rendszerbe:
  - a rendszer elveszíti nyelvfüggetlenségét (gazdaságossági következmények)
  - a nyelvi többértelműségek minden szinten megjelennek (ez túlgeneráláshoz vagy hibás találatok tömegéhez vezethet)
  - mind a feldolgozás, mind pedig a keresés idő- és tárigénye többszörösére nő
- A jelenleg angol nyelvre rendelkezésre álló nyelvtanunk túl bonyolult és nagy ahhoz képest, hogy célunk csupán a főnévi szerkezetek és a mondatváz előállítása, sőt, a nyelvészek véleménye sem egységes a tekintetben, hogy mit is nevezhetünk főnévi szerkezetnek. Rendszerünkben ezért egyelőre az emberi fordító feladata lesz a főnévi szerkezetek kijelölése.
- A rendszer nem kezeli az egyes nyelvi szerkezeteket alkotó szavak nyelvi „jelentőségét", azaz ugyanannyira bünteti egy határozószó hiányát, mint az igei szerkezet fejét képező ige különbözőségét, vagy a főnévi szerkezetek esetén egy melléknév különbségét és a szerkezet fejének különbözőségét (pl. a „száraz pezsgő"-től ugyanakkor távolságra van a „félédes pezsgő", mint a „száraz pelenka"). Ennek a megoldása egy későbbi időben várható.
- A keresést meggyorsító indexelési technika kifejlesztése a következő feladat. A rendszer használhatóságának elengedhetetlen feltétele, hogy hatékony indexelési technika támogassa.

7 Összefoglalás

A cikkben bemutattam a nyelvi hasonlóság szerinti intelligens keresésre adott eddigi elméleti megoldásomat, amely a MorphoLogic MetaMorphoTM rendszerébe illeszkedően kerül majd tesztelésre és felhasználásra. Definíciót adtam a több szintű nyelvi hasonlóság mértékére és az ez alapján számítható hasonlóság-vektorra. Valamint vázoltam a közel- és távolabbi jövő kutatási és fejlesztési feladatait.

Irodalomjegyzék

[Levenshtein65] Levenshtein, V. I.: 'Binary codes capable of correcting deletions, insertions and reversals', (1965) Doklady Akademii Nauk, SSSR 163(4) p845-848, also Soviet Physics Doklady 10(8) p707-710.

[Mandreoli02] Mandreoli, F, Martoglia R., and Tiberio, P. (2002) Searching Similar (Sub)Sentences for Example-Based Machine Translation. *Atti del Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati (SEBD 2002)*, Isola d'Elba, Italy.

[Nagao84] Nagao, M.: 'A framework of a mechanical translation between Japanese and English by analogy principle', In A. Elithorn and R. Banerji (eds.) (1984), Artificial and human intelligence, 173-180. Amsterdam: North-Holland.

[Navarro01] Navarro, G. 'A Guided Tour to Approximate String Matching.' (2001) ACM Computing Surveys, 33(1):31-88.

[Navarro01] Navarro, G., Baeza-Yates, R., Sutinen, E., Tarhio, J. 'Indexing Methods for Approximate String Matching', (2001) IEEE Data Engineering Bulletin, 24(4), 19--27, Special issue on Managing Text Natively and in DBMSs.

[Planas00] Planas, E., Furuse: O. 'Multi-Level Similar Segment Matching Algorithm for Translation Memories and Example-Based Machine Translation' (2000) COLING-2000, Saarbruecken, Germany, 621-627.

[Prószéky02] Prószéky, G. and L. Tihanyi: 'MetaMorpho: A Pattern-Based Machine Translation Project'. (2002) Translating and the Computer 24, ASLIB, London.

[Prószéky96] Prószéky 'Syntax As Meta-morphology', (1996) Proceedings of COLING-96, Vol.2, 1123–1126. Copenhagen, Denmark.

[Wagner& Fischer74] Wagner, A. R., Fischer M. (1974) The String-to-string Correction Problem. Journal of the ACM, Vol. 21, #1, pp. 168-173.